

Analisis y Diseño de Algoritmos

Juan Miguel Rojas

29 de mayo de 2024

Understanding Serialization in Sequential Machines

Explore and understand the concept of serialization in converting concurrent processes to sequential operations within a sequential machine, using a specific algorithm as a case study.

Serialización en el documento

La serialización es un procedimiento que tiene como objetivo simular correctamente y completamente una relación síncrona a partir de una relación asíncrona. Esto basicamente consiste en convertir un objeto o estructura de datos en un formato el cual pueda ser almacenado y transmitido.

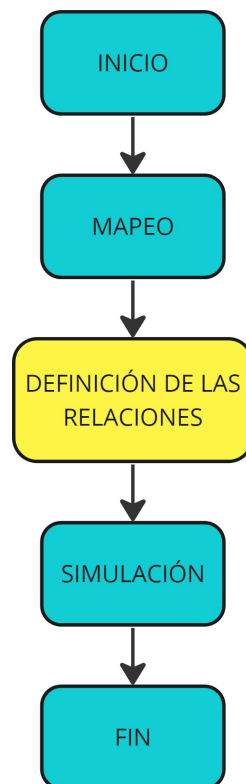


Figura 1: Diagrama

Se menciona un ejemplo concreto en el documento donde se compara el modelo híbrido con el modelo DeGroot. La aplicación de reglas específicas en el modelo híbrido puede resultar en un gran número de estados sucesores posibles, lo que hace que la verificación de la existencia de un estado alcanzable que cumpla con el consenso sea compleja.

Detalles del procedimiento de serialización

- Se implementa un lenguaje de programación llamado Maude la cual permite comprender como se moldean las opiniones.
- Con esto, podemos aplicar tecnicas formales como el analisis de alcanzabilidad para poder ver la polarización en redes sociales.
- La serialización se aplica a conjuntos de relaciones asíncronicas, paralelas y sincronas como cierres de una de una relación de un conjunto atomico.

Procedimiento del algoritmo

El algoritmo utiliza estrategias de Maude, como mencionamos anteriormente, para verificar la existencia de un estado alcanzable que cumpla con las condiciones de un sistema dado. Tenemos las siguientes definiciones claves:

- **Relación asincronica**
- **Relación Paralela**
- **Relación Sincrónica**

Definimos un nuevo conjunto $T\cdot$ y una biyección $mark$ que mapea los terminos en T a $T\cdot$, luego se define la relación $\rightarrow \cdot$ en $U\odot$, tal que, $A \rightarrow \cdot C \iff A \rightarrow B$ y $C = mark(B)$. Luego se define la relación $R \rightarrow S$ en U tal que $\langle A; B \rangle \in R \rightarrow S \iff B = (A \setminus A') \cup mark - 1(B')$.

Este procedimiento de serialización permite simular las relaciones sincronicas a partir de relaciones asincronicas, garantizando correctitud y completitud.

Aplicaciones del algoritmo

- Facilitar la ejecución y operacionalización de semántica en lenguajes de programación.
- Optimizar la implementación de reglas de reescritura en sistemas de computación.
- Facilitar la comprensión y el análisis de relaciones entre conjuntos de datos en sistemas informáticos.

Especificación del problema

Entrada

- Un conjunto finito de términos T , representados por a, b y c.
- Se utilizan relaciones de conjuntos concurrentes, redices, estrategias de reducción y funciones de actualización de opiniones.

Salida

- Se generan nuevos estados del sistema con las opiniones actualizadas de los agentes, lo que representa la evolución dinámica de las opiniones en el contexto del sistema.

Tecnicas de programación

Realmente no encuentre una tecnica de programación en especifico, esto debido a que no conozco mas tecnicas de programación y de las vistas en el curso no logre encontrar una tecnica que se evidenciara correctamente en el modelo.

Complejidad

El analisis de complejidad fue de $O(n \cdot p)$ que estas denontan las relaciones de los conjuntos

Aplicación de De Groot

- En un ejemplo específico, la serialización se utiliza para actualizar las opiniones de agentes en función de las influencias recibidas.
- Las actualizaciones muestran cómo las opiniones de los agentes pueden cambiar con cada nueva influencia considerada.
- Todo este proceso nos muestra como las opiniones llegan a evolucionar a lo largo del tiempo debido a diferentes interacciones y contextos.

References

[1] Rocha, C., Ramírez, C., Dowek, G. (2011). A formal library of set relations and its application to synchronous languages.

[2] Rocha, C., Muñoz, C., Olarte, C., Valencia, F. (2024). Unified opinion dynamic modeling as concurrent set relations in rewriting logic.